The Beltrami Flow over Manifolds

Nir Sochen, Rachid Deriche, Lucero Lopez Perez

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

The Beltrami Flow over Manifolds

Nir Sochen , Rachid Deriche , Lucero-Lopez Perez

N° 4897

July 2003

THÈME 3

ISSN 0249-6399 ISRN INRIA/RR--4897--FR+ENG

The Beltrami Flow over Manifolds

Nir Sochen *, Rachid Deriche †, Lucero-Lopez Perez †

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Odyssée

Rapport de recherche n° 4897 — July 2003 — 35 pages

Abstract: In many medical computer vision tasks, the relevant data is attached to a specific tissue such as the cortex or the colon. This situation calls for regularization techniques which are defined over non flat surfaces. We introduce in this paper the Beltrami flow over manifolds. This new regularization technique overcomes the over-smoothing of the L_2 flow and the staircasing effects of the L_1 flow, that were recently suggested via the harmonic map methods. The key of our approach is first to clarify the link between the intrinsic *Polyakov action* and the implicit *Harmonic energy functional* and then use the geometrical understanding of the Beltrami Flow to generalize it to images on explicitly and implicitly defined non flat surfaces. It is shown that once again the Beltrami flow interpolates between the L_2 and L_1 flows on non-flat surfaces. The implementation scheme of this flow is presented and various experimental results obtained on a set of various real images illustrate the performances of the approach as well as the differences with the harmonic map flows. This extension of the Beltrami flow to the case of non flat surfaces opens new perspectives in the regularization of noisy data defined on manifolds.

Key-words: Regularization on manifolds, Beltrami Flow, Polyakov action, Harmonic maps, Scalar and Color image regularization, Non flat surfaces.

* Dept. of Applied Mathematics,University of Tel-Aviv,Ramat-Aviv, tel-Aviv 69978,Israel, sochen@math.tau.ac.il, www.math.tau.ac.il/ sochen/

† Projet Odyssée, INRIA Sophia-Antipolis, 2004, route des Lucioles, BP 93, 06902 Sophia-Antipolis, France, {Rachid.Deriche,Lucero.Lopez_perez}@sophia.inria.fr, www-sop.inria.fr/odyssee/presentation/index.en.html

Flot de Beltrami sur des Variétés

Résumé : Ce rapport traite du problème de la régularisation de données bruitées définies sur des surfaces non planes et représentées de manière implicite ou intrinsèque. Deux classes d'approches existent à ce jour pour la régularisation de données bruitées définies sur des variétés : L'approche que l'on appelle *Action de Polyakov*, utilisée dans le développement du flot de Beltrami et qui nécessite une représentation locale et intrinsèque de la surface de travail, et l'approche dite des *Cartes Harmoniques*, qui utilise une représentation implicite de la surface où vivent les données à régulariser. Dans ce rapport, on commence par clarifier le lien qui existe entre l'*Action de Polyakov*, intrinsèque, et l'approche implicite des *cartes harmoniques*. Le flot de Beltrami qui agit comme un opérateur de régularisation qui interpole, dans le cas des images scalaires, entre la norme L_2 et la norme L_1 est généralisé dans ce rapport au cas des images définies sur des surfaces non planes. Nous développons dans ce rapport de nouveaux flots de régularisation de données scalaires ou vectorielles définies sur des variétés représentées de manière implicite ou explicite. On dérive en particulier les flots qui permettent de régulariser des images scalaires et couleurs définies sur des surfaces représentées de manière intrinsèque et on explicite ces flots dans le cas de surfaces sphériques. Pour le cas des surfaces représentées de manière implicite, on propose deux approches, une géométrique et l'autre variationnelle, pour régulariser des données scalaires définies sur des courbes ou des surfaces représentées de manière implicite. On montre que là aussi, le flot de Beltrami sur des surfaces non planes interpole entre la norme L_2 et la norme L_1. Nous pensons, comme le démontre notre application sur la régularisation de cartes rétinotopiques sur la surface corticale, que les travaux présentés dans ce rapport permettent d'ouvrir de nouvelles perspectives dans le traitement de données vectorielles sur des surfaces non planes.

Mots-clés : Régularisation de données sur des variétés, Flot de Beltrami., Action de Polyakov, Cartes harmoniques, Régularisation de données scalaires et vectorielles sur des images non planes, Surface corticale.

1 Introduction

We have seen in recent years an expansion in the use of differential geometry and calculus of variation for various problems in image processing, computer vision and computer graphics. In particular, one can notice that the problem of regularizing noisy data defined on non flat, implicit or intrinsic, surfaces has been tackled with two functionals which operate on the space of embedding maps of Riemannian manifolds (see [8] for non-variational approach)

- **The Polyakov action:** It was introduced in the Beltrami framework [19, 20, 22, 9, 10]. It uses a local and intrinsic-parametric description of the manifolds and treat the metric as a dynamic variable.
- **The Harmonic maps**. Recently used to denoise images on manifolds [1, 13, 5, 2, 17]. It uses an implicit description for the surface where the noisy data are constrained to live. The metric is a parameter of the process.

It is the aim of this paper to first clarify the relation between the intrinsic Polyakov action of the Beltrami framework [19, 20, 22, 9, 10] and the implicit harmonic energy functional [1, 13, 5, 2, 17] . It is found that although the functionals are basically the same, there are differences in the way various problems are formulated and consequently in the way the functionals are applied. Specifically for the case of denoising images on non-flat surfaces there are differences in the definition of the manifolds and of the embedding functions.
Usually we have, in various problems in vision, an underline manifold, flat or curved, and the features are defined over this manifold. A typical situation is when the image rectangle is the underlying flat manifold and at each pixel we assign values such as intensity, color, gradient value, gradient direction, motion vector, disparity vector, texture characteristics etc. This is easily described mathematically as a **fiber bundle** in which a space is attached to each point in the base manifold. The spaces at different points of the manifolds are isomorphic. A choice of one point in the attached space for every point in the base manifold is called a **section**. If the attached space is a vector space then this section is called a vector field.
In order to understand better the difference between the two formulations, we consider the case of a gray-value image defined on a surface (flat or not). In the harmonic energy approach it is usually assumed that the map is from the 2D surface (the base manifold) to the real axis (the fiber). This means that **the metric of the base manifold** is used for the derivatives and the fiber's 1D flat metric is used for the values of the scalar field. If the norm chosen is L_2 we get a linear diffusion process

whether the base manifold is flat or not. In the case of L_1 norm or the Φ formulation [11, 16, 7] we get non-linear flows. These flows only depend on the absolute value of the gradient, defined over the base manifold [2, 23].
In the Beltrami framework [19, 20, 9], the basic object is **the section embedded in the fiber bundle**. For a flat gray-value image, thus, the graph of the intensity function is the section of this 3D fiber bundle and is the primary object of interest. The metric of the fiber bundle induces a metric on this section and both are used in the functional. The flow depends on the **geometry of the data** and not only on the geometry of the base manifold. This means that **the image metric, that is, the section metric is being changed along the flow**. It also means that the flow may depend on the direction of the gradients and not only on their amplitudes. The Beltrami flow which has been shown to interpolate, for flat gray-value images, between the L_2 norm and the L_1 norm [20, 21] is generalized in this paper to images on non-flat surfaces. It is shown, in this work, that once again the Beltrami flow interpolates between the L_2 and L_1 flows on non-flat surfaces that were derived recently [2, 23]. A common feature of both frameworks is the ability to deal with non-flat feature spaces [22, 23, 10, 24, 3, 4].
This report is organized as follows: In Section 2 we briefly review the Beltrami and harmonic map frameworks. The intrinsic-implicit correspondence is explained in Section 3. The Beltrami intrinsic approach is used in Section 4 to derive equations for the denoising of gray value images defined over an explicitly described non-flat 2D surfaces. A general, as well as an example of an image on the sphere are worked out explicitly. The straight forward generalization for color images is described in Section 5. We take in Section 6 the Beltrami formulation of scalar denoising of data on a curve and reformulate it in an implicit form. The equations are derived first from pure geometric considerations. It is shown next that the derived implicit equation is the gradient descent equation for a functional. Section 7 generalize these results to a scalar field defined over an implicit manifold. The interpolation between the L_1 and L_2 norms is discussed in Section 8. The regularization of a vectorial field over an implicit surface is discussed in Section 9. Section 10 presents examples and results. We summarize and conclude in Section 11.

2 The Beltrami and the Harmonic Map Frameworks

There are two ways to write the geometric functional which is called harmonic energy or the Polyakov action. Both functionals are defined over the space of embedding maps of Riemannian manifolds. We focus our attention to the case of a gray-value

image defined over a two- (or one-) dimensional manifold i.e. a surface (curve). The first way is to represent the surface implicitly i.e. as a zero section of a distance function, defined over $\mathbb{R}^3$. The other way will be to choose a local coordinate system and define the surface intrinsically in a parametric way. These ideas are developed in more details in the next two-subsections.

2.1 Implicit formulation

Let's suppose that we have a scalar data U defined on a known surface Σ, that is we have a mapping $\tilde{U} : \Sigma \rightarrow \mathbb{R}^1$ where Σ is a known surface, given by its implicit representation

$$\Psi(X^1, X^2, X^3) = 0 \ . \tag{1}$$

We extend $\tilde{U}$ to $U : \mathbb{R}^3 \rightarrow \mathbb{R}$ such that $\tilde{U}$ is the restriction of U to Σ. The *Harmonic maps* approach [1, 13, 5, 2] is to look for the minimizer of the following energy

$$S_{imp}[U] = \int_{\mathbb{R}^3} || \ \nabla_\Sigma U \ ||^2 \ d\Sigma \tag{2}$$

where $d\Sigma$ is the surface element $d\Sigma = \delta(\Psi)||\nabla\Psi||dX^1dX^2dX^3$. We denote here by δ the Dirac function and $\nabla_\Sigma U$ denotes the gradient intrinsic to the surface Σ i.e the projection of the 3D Cartesian gradient of U on $T\Sigma$ – the tangent space to Σ. Clearly $\nabla\Psi$ is normal to its zero level set, which is Σ. It follows then that $\nabla_\Sigma U = P_\eta \nabla U$ where P_η is the orthogonal projection operator on the surface Σ. Here ∇U is the 3D Cartesian gradient of U:

$$\nabla U = \left(\frac{\partial U}{\partial X^1}, \frac{\partial U}{\partial X^2}, \frac{\partial U}{\partial X^3} \right)^t \ ,$$

and η denotes the unit normal vector to the surface Σ with

$$\eta = \frac{\nabla\Psi}{||\nabla\Psi||}.$$

We can rewrite now the harmonic energy as follows :

$$\begin{aligned} S_{imp}[U] &= \int_{\mathbb{R}^3} || \ P_\eta \nabla U \ ||^2 \ d\Sigma \\ &= \int_{\mathbb{R}^3} || \ P_\eta \nabla U \ ||^2 \ \delta(\Psi) \ || \ \nabla\Psi \ || \ dX^1 dX^2 dX^3 \ . \end{aligned}$$

All the expressions above are considered in the sense of distributions. Noting that given a surface with a unit normal vector η, the orthogonal projection operator P_η on that surface is given by :

$$P_\eta = I - \eta\eta^t \ , \tag{3}$$

where I is the 3×3 Identity matrix and η^t denotes the transposed of the vector η. The functional can be rewritten as follows :

$$S_{imp}[U] = \int_{\mathbb{R}^3} \nabla U^t (I - \eta\eta^t) \nabla U d\Sigma \ . \tag{4}$$

This functional is the generalization of the L_2 norm from flat to non-flat manifolds. The modified gradient descent equation reads:

$$U_t = \frac{1}{||\nabla\Psi||} \mathrm{Div} \left(||\nabla\Psi|| P_\eta \nabla U \right) \ , \tag{5}$$

where Div stands for the 3D Divergence operator, and ∇ is in $\mathbb{R}^3$ as well. The restriction of U to the surface Σ is the regularized image. The implicit functional can be generalized via the Φ formulation [11, 16, 7] to

$$S_{imp}[U] = \int_{\mathbb{R}^3} \Phi \left(||\nabla_\Sigma U|| \right) d\Sigma \ , \tag{6}$$

and in particular, for the generalized L_1 norm

$$S_{imp}[U] = \int_{\mathbb{R}^3} ||\nabla_\Sigma U|| d\Sigma \tag{7}$$

we obtain the following minimization flow

$$U_t = \frac{1}{||\nabla\Psi|| ||P_\eta \nabla U||} \mathrm{Div} \left(\frac{||\nabla\Psi|| P_\eta \nabla U}{||P_\eta \nabla U||} \right) \ . \tag{8}$$

For the interested reader, we refer to [1, 13, 5, 2] for more details and results on this implicit formulation and its various applications.

2.2 Intrinsic formulation

Suppose we have a 2-dimensional manifold Σ with local coordinates σ^1, σ^2 embedded in an 3-dimensional manifold M with coordinates X^1, X^2, X^3, the embedding map $X : \Sigma \to M$ is given explicitly by the 3 functions of 2 variables

$$X : (\sigma^1, \sigma^2) \longrightarrow (X^1(\sigma^1, \sigma^2), X^2(\sigma^1, \sigma^2), X^3(\sigma^1, \sigma^2)) \ .$$

Note that the X^is are the solution of Eq. (1). A gray-value image is represented, in this framework, as the embedding $(X^1 = \sigma^1, X^2 = \sigma^2, X^3 = \beta U(\sigma^1, \sigma^2))$ (see Fig. 1).

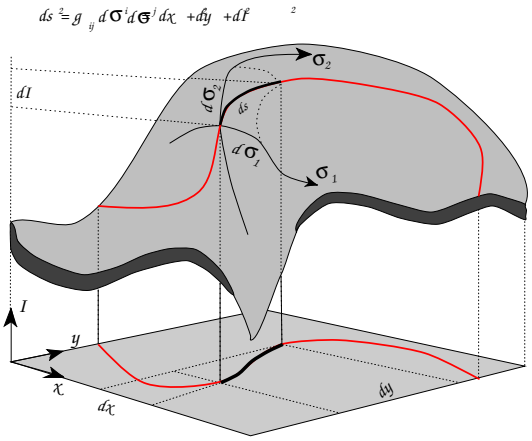


Figure 1: The graph of the intensity map as a surface embedded in $\mathbb{R}^3$

Denote by (Σ, g) the image manifold and its metric and by (M, h) the space-feature manifold and its metric, then the map $X : \Sigma \to M$ has the following weight :

$$S_{int}[U, g] = \int d\sigma^1 d\sigma^2 \sqrt{g} g^{\mu\nu} \partial_\mu X^i \partial_\nu X^j h_{ij}(X) \tag{9}$$

where g is the determinant of the image metric, $g^{\mu\nu}$ is the inverse of the image metric, the range of indices is $\mu, \nu = 1, .., 2$ and $i, j = 1, ..3$ and h_{ij} is the metric of the embedding space. We use the summation convention: indices that appear twice are being summed over. This functional was first proposed by Polyakov in the context of high energy physics [15] .
Let us formulate the Polyakov action in matrix form: (Σ, G) is the image manifold and its metric as before. Similarly, (M, H) is the spatial-feature manifold and its metric (it is simply $\mathbb{R}$ with the usual Euclidean distance for gray-value images). Define

$$A^{ij} = (\partial_\Sigma X^i)^t G^{-1} \partial_\Sigma X^j \ ,$$

where $\partial_\Sigma X^i = (\partial_{\sigma^1} X^i, \partial_{\sigma^2} X^i)^t$.

The map $X : \Sigma \to M$ has a weight

$$S[X^i, G, H] = \int dA \ \mathrm{Tr}(AH),$$

where $g = \det(G)$, and the area element $dA = d\sigma^1 d\sigma^2 \sqrt{g}$.
The gradient descent equations, with respect to the embedding functions, are

$$X^i_t = \frac{1}{\sqrt{g}} \partial_\mu (\sqrt{g} g^{\mu\nu} \partial_\nu X^i) + \Gamma^i_{jk} (\partial_\mu X^j)(\partial_\nu X^k) g^{\mu\nu}. \tag{10}$$

In our case X^1 and X^2 are fixed and we only change $X^3 = \beta U$. The embedding space is Euclidean and therefore the Levi-Civita Γ^3_{jk} coefficients are zero.
The flow, in matrix form, is therefore

$$U_t = -\frac{1}{2\sqrt{g}} \frac{\delta S}{\delta U} = \underbrace{\frac{1}{\sqrt{g}} \mathrm{div} \left(\sqrt{g} G^{-1} \nabla U \right)}_{\Delta_g U}. \tag{11}$$

The extension for non-Euclidean embedding space is treated in [19, 20, 9, 21, 22]. Note that we didn't specify the metric yet. If we choose for a metric the first fundamental form of the base, non-flat, manifold then this flow is the analog of Eq. (5). Both equations describe a **linear operator** that acts on the image function U. One major difference between the Beltrami framework and the harmonic map formulation is the way one choose the metric. In the harmonic map approach it is treated as data which is given beforehand. The metric then is **constant in time**. In the Beltrami framework **the metric is a dynamic variable to be found by minimizing the functional**. Carrying out the calculation we find the following equation (see [19] for the derivation):

$$\frac{1}{\sqrt{g}} \frac{\delta S}{\delta g_{\mu\nu}} = -\frac{1}{2} g_{\mu\nu} g^{\lambda\rho} \partial_\lambda X^i \partial_\rho X^j h_{ij} + \partial_\mu X^i \partial_\nu X^j h_{ij} = 0 \ , \tag{12}$$

Albeit the complicated appearance of the minimization equation **it can be solved analytically**. The result is simply

$$g_{\mu\nu} = \partial_\mu X^i \partial_\nu X^j h_{ij} \ .$$

This equation has a clear geometric meaning: It is the induced metric of the section! In the simplest situation of flat domain with a scalar field i.e. the intensity, it is find

to be [20] :

$$(g_{\mu\nu}) = \begin{pmatrix} 1+U_x^2 & U_xU_y \\ U_xU_y & 1+U_y^2 \end{pmatrix}. \tag{13}$$

where U denotes the intensity level. In the Beltrami framework, therefore, the manifold of interest, and the metric, is taken to be **the data surface** i.e. the graph of the intensity function. In other words we are interested in the structure of the section of the fiber bundle and not in the base manifold. It means that the metric itself depends on the data. This is the way non-linearities are introduced in the flow. Equation (11), with the induced metric Eq. (13), has a clear geometric meaning. It is the projection of the mean curvature vector in the $X^3 = U$ direction:

$$U_t = KN_{\hat{U}} \tag{14}$$

where K is the mean curvature magnitude. N is the normal to the surface and $\hat{U}$ is a unit vector in the $X^3 = U$ direction (see Fig. 2). This choice of the induced metric gives a simple geometric meaning to the functional as well. It becomes simply the volume of the section (area or length for the two- and one- dimensional cases). It

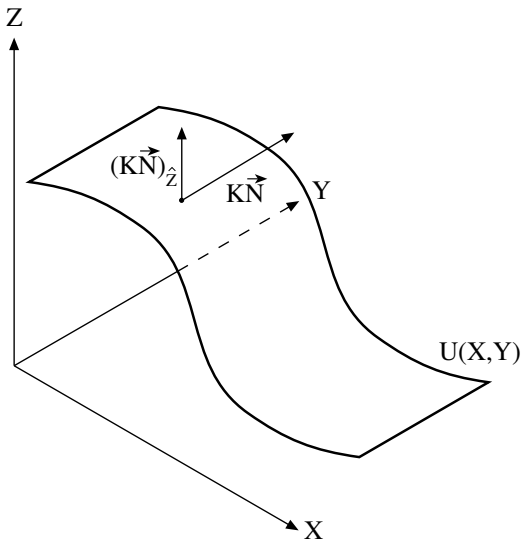


Figure 2: The Beltrami flow velocity as a projection of the mean curvature on the data = z axis

is important for the study below to have the explicit form of the functionals for flat domain in the scalar and vectorial cases:

$$\begin{aligned} S_{\text{scalar}}[U] &= \int dxdy\sqrt{g} = \int dxdy\sqrt{1+\beta^2||\nabla U||^2} \\ S_{\text{vector}}[U^i] &= \int dxdy\sqrt{1+\beta^2\sum_i||\nabla U^i||^2+\beta^4\sum_{i,j}||\nabla U^i\times\nabla U^j||^2}\ , \end{aligned} \quad (15)$$

where β is the ratio between the distances taken in the spatial and intensity directions.
We will use this geometric understanding in order to derive the Beltrami flow on non-flat surfaces from pure geometric consideration. We will also use this simplified functional in our quest for implicit formulation of the Beltrami flow on non-flat manifolds. But before we turn to that let us explain first the implicit-intrinsic correspondence.

3 The Implicit-Intrinsic Correspondence

We have shown two ways to write the geometric functional, which is called harmonic energy or the Beltrami functional, that weight embedding maps of Riemannian manifolds.
Let us work out the case of a painted surface. The mapping is $U:\Sigma\rightarrow\mathbb{R}^1$ where Σ is a known surface with metric (= first fundamental form) G.
One way is to represent the surface implicitly i.e. as a zero section of an Osher-Sethian function, defined over $\mathbb{R}^3$. The other way will be to choose a local coordinate system and define the surface intrinsically in a parametric way.
The implicit method consist on the equation

$$\Psi(X^1,X^2,X^3)=0. \quad (16)$$

This equation is an implicit description of a surface. The functional is written in this case as

$$S_{imp}[U]=\int_{\mathbb{R}^3}\nabla U^t(\mathrm{I}-\eta\eta^t)\nabla U\delta(\Psi)||\nabla\Psi||dX^1dX^2dX^3 \quad (17)$$

where

$$\eta=\frac{\nabla\Psi}{||\nabla\Psi||}$$

and the gradient is

$$\nabla U = \left(\frac{\partial U}{\partial X^1}, \frac{\partial U}{\partial X^2}, \frac{\partial U}{\partial X^3} \right)^t$$

The intrinsic way consists on choosing (σ^1, σ^2) as local coordinates on (possibly part of) the manifold. The surface is given then by the functions

$$\left(X^1(\sigma^1, \sigma^2), X^2(\sigma^1, \sigma^2), X^3(\sigma^1, \sigma^2) \right). \tag{18}$$

Note that these functions are the solutions of Eq. (16).
The functional is defined as

$$S_{int}[U] = \int \partial_\Sigma U^t G^{-1} \partial_\Sigma U \sqrt{g} d\sigma^1 d\sigma^2 \tag{19}$$

where $g = \det(G)$ and

$$\partial_\Sigma U = \left(\frac{\partial U}{\partial \sigma^1}, \frac{\partial U}{\partial \sigma^2} \right)^t$$

We will show the equality between these two ways of writing by passing from the implicit to the intrinsic formulation. In order to do that we will perform the integration over the delta function and express all the the three dimensional objects intrinsically. Explicitly we prove the following technical lemma:
LEMMA:

1. $$G = J^t J$$

   where J is the Jacobian – a 3×2 matrix whose elements are

   $$\frac{\partial X^i}{\partial \sigma^\mu} \qquad i = 1, 2, 3 \quad \mu = 1, 2$$

   That is

   $$J = \begin{pmatrix} \frac{\partial X^1}{\partial \sigma^1} & \frac{\partial X^1}{\partial \sigma^2} \\ \frac{\partial X^2}{\partial \sigma^1} & \frac{\partial X^2}{\partial \sigma^2} \\ \frac{\partial X^3}{\partial \sigma^1} & \frac{\partial X^3}{\partial \sigma^2} \end{pmatrix}$$

2. $$\nabla U^t J = \partial_\Sigma U^t$$

3.
$$||\nabla\Psi|| = \sqrt{g}$$
where $g = \det G$.

4.
$$J^t \cdot \nabla\Psi = 0$$

PROOF:

1. For an intrinsic surface the induced metric is
$$G_{\mu\nu} = e_\mu \cdot e_\nu$$
where e_1, e_2 are the tangent vectors to the surface. In our case
$$e_\mu = \begin{pmatrix} \frac{\partial X^1}{\partial \sigma^\mu} \\ \frac{\partial X^2}{\partial \sigma^\mu} \\ \frac{\partial X^3}{\partial \sigma^\mu} \end{pmatrix}$$
The Jacobian is simply written as $J = (e_1, e_2)$ and we find
$$(J^t J)_{\mu\nu} = e_\mu \cdot e_\nu = G_{\mu\nu}$$

2.
$$\partial_{\sigma^\mu} U = U_{x^1}\frac{\partial X^1}{\partial \sigma^\mu} + U_{x^2}\frac{\partial X^2}{\partial \sigma^\mu} + U_{x^3}\frac{\partial X^3}{\partial \sigma^\mu} = \nabla U^t e_\mu \qquad \mu = 1, 2$$
This is written as
$$\partial_\Sigma U^t = \nabla U^t J.$$

3. The normal to the surface $\nabla\Psi$ is perpendicular to the two tangent vectors e_μ. The components of vector product of two vectors V_1 and V_2 is given by $(V_1 \times V_2)_i = \epsilon_{ijk} V_1^j V_2^k$ where $\epsilon_{ijk} = (-1)^p$ and p is the number of permutations needed to bring (ijk) to (123). We can now compute
$$||\nabla\Psi||^2 = ||e_1 \times e_2||^2 = \epsilon_{ijk} e_1^j e_2^k \epsilon_{ilm} e_1^l e_2^m \ .$$
We use the identity $\epsilon_{ijk}\epsilon_{ilm} = \delta_{jl}\delta_{km} - \delta_{jm}\delta_{kl}$ which can be verified directly, and find
$$||\nabla\Psi||^2 = (\delta_{jl}\delta_{km} - \delta_{jm}\delta kl) e_1^j e_2^k e_1^l e_2^m = e_1 \cdot e_1 e_2 \cdot e_2 - (e_1 \cdot e_2)^2 = G_{11}G_{22} - G_{12}^2 = \det G$$

4.
$$J^t \cdot \nabla\Psi = (e_1, e_2)^t \nabla\Psi$$
but
$$e_i^t \cdot \nabla\Psi = \partial_{\sigma^i}\Psi = 0 \quad i = 1, 2,$$
Since Ψ is constant in the tangent directions. That concludes the proof of the Lemma.

Inserting these results in Eq. (19) we find

$$S_{int}[U] = \int \nabla U^t J G^{-1} J^t \nabla U \delta(\Psi) ||\nabla\Psi|| dX^1 dX^2 dX^3 \tag{20}$$

All is needed now is to prove the identity

$$JG^{-1}J^t = Id - \eta\eta^t$$

In order to do that we start from property 1 in the Lemma: $G = J^t J$. We take now the inverse of the two sides. Here we must proceed with care. While G is positive definite and there is no problem, J has a non-trivial kernel as we can see from property 4 of the Lemma. The Kernel is one dimensional and is the normal direction to the surface. Its inversion should be accompanied by projection on the tangent space of the surface where J^{-1} is well defined. We get therefore

$$G^{-1} = (J^t J)^{-1} = J^{-1}(Id - \eta\eta^t)^t (Id - \eta\eta^t) J^{-t} = J^{-1}(Id - \eta\eta^t) J^{-t}$$

and the identity follows.

4 Scalar field on a surface: Explicit-intrinsic approach

Assume that we have a surface Σ, and a Riemannian structure on Σ i.e a metric $(\tilde{g}_{ij})$. We can take, for example, the induced metric of Σ as in the previous Section. This surface is the substance on which an image is defined. The regularization acts on the image only and do not change the geometry of the surface. The metric $(\tilde{g}_{ij})$ is constant with respect to time. In addition we are given a scalar field (image) i.e. a function U, defined on Σ. In a local coordinate system (x, y) of the surface Σ the line element is

$$ds_\Sigma = \tilde{g}_{11} dx^2 + 2\tilde{g}_{12} dx dy + \tilde{g}_{22} dy^2$$

We add now a third dimension, perpendicular to the surface and describe the scalar field $U(x,y)$ as a surface S embedded in the 3D fiber bundle $\mathcal{M} = \Sigma \times \mathbb{R}$. The surface S is a section of the fiber bundle. The embedding coordinates are

$$\left(x(\sigma^1,\sigma^2) = \sigma^1, y(\sigma^1,\sigma^2) = \sigma^2, X^3(\sigma^1,\sigma^2) = U(x,y)\right)$$

or by abuse of notations: $(x,y,U(x,y))$, where (x,y) are coordinates on S .
The line element in the manifold/fiber bundle $\mathcal{M}$ is

$$ds_{\mathcal{M}} = \tilde{g}_{11}dx^2 + 2\tilde{g}_{12}dxdy + \tilde{g}_{22}dy^2 + \beta^2 dU^2$$

or, in equivalent way, the metric of the 3D manifold $\mathcal{M}$ is

$$(h_{ij}) = \begin{pmatrix} \tilde{g}_{11} & \tilde{g}_{12} & 0 \\ \tilde{g}_{21} & \tilde{g}_{22} & 0 \\ 0 & 0 & \beta^2 \end{pmatrix}.$$

Assuming an isometric embedding i.e. $ds_{\mathcal{S}} = ds_{\mathcal{M}}$, we obtain the induced metric on the section S :

$$ds_{\mathcal{S}} = \tilde{g}_{11}dx^2 + 2\tilde{g}_{12}dxdy + \tilde{g}_{22}dy^2 + \beta^2\left(U_x dx^2 + 2U_x U_y dxdy + U_y^2 dy^2\right) \ ,$$

where β^2 takes into account the differences in dimensions between the spatial directions and the intensity one. The metric elements on S , therefore, are:

$$g_{ij} = \tilde{g}_{ij} + \beta^2 U_i U_j \ . \tag{21}$$

Denote by G the matrix whose elements are g_{ij}. Using this metric we can calculate the Laplace-Beltrami operator:

$$\Delta_g U = \frac{1}{\sqrt{g}}\mathrm{Div}\left(\sqrt{g}G^{-1}\nabla U\right) \ . \tag{22}$$

Note that here Div and ∇ are two-dimensional. The equation of motion that results from the Polyakov action is:

$$U_t = \Delta_g U + \Gamma^3_{ab}(\partial_i X^a)(\partial_j X^b)g^{ij} \ ,$$

where

$$\Gamma^c_{ab} = \frac{1}{2}h^{cd}\left(\partial_a h_{db} + \partial_b h_{ad} - \partial_d h_{ab}\right) \ .$$

In our case

$$\Gamma^3_{ab} = \frac{1}{2}h^{3d}\left(\partial_a h_{db} + \partial_b h_{ad} - \partial_d h_{ab}\right) = \frac{1}{2}h^{33}\left(\partial_a h_{3b} + \partial_b h_{a3} - \partial_3 h_{ab}\right) = 0 \ .$$

where the second equality is from the decoupling of the third dimension from the other two (i.e. the independence of the fiber on the base manifold) and the last equality is from the fact that $h_{33} = const$ and the other elements do not depend on U (note that $\partial_3 \equiv \partial/\partial X^3 = \partial/\partial U$) .
The equation of motion is therefore

$$U_t = \Delta_g U \ ,$$

where $\Delta_g U$ is given in Eq. (22) with the induced metric that we calculated Eq. (21).

4.1 Example: Gray image on the sphere

In order to apply the above equations to an image "painted" on the sphere we have to calculate the metric $\tilde{g}_{ij}$ of the sphere. We choose to represent the sphere by stereographic coordinates from the south pole. The metric in this case is

$$\tilde{g}_{ij} = \frac{4}{1 + x^2 + y^2}\delta_{ij} \ .$$

See for example ([18, 10]).
In general we write

$$g_{ij} = \tilde{g}_{ij} + \beta^2 U_i U_j \ ,$$

where $U_i = \partial U/\partial x^i$. We denoted here $x^1 = x$ and $x^2 = y$. Define

$$A(x, y) = \frac{4}{1 + x^2 + y^2} \ .$$

We find, consequently, that

$$g = det(g_{ij}) = A^2 + A\beta^2(U_x^2 + U_y^2) \ ,$$

and the equation of motion is

$$U_t = \frac{1}{\sqrt{g}}\mathrm{Div}\left(\frac{1}{\sqrt{g}}\begin{pmatrix} A + \beta^2 U_y^2 & -\beta^2 U_x U_y \\ -\beta^2 U_x U_y & A + \beta^2 U_x^2 \end{pmatrix}\nabla U\right) \ .$$

5 Color image on a surface: Explicit-intrinsic approach

The second case, in levels of complexity, is a color image on a surface. Here the base manifold is a two-dimensional surface. The fiber is the color space. We will treat here, for simplicity, the case where the color space is taken to be a Euclidean $\mathbb{R}^3$ space with Cartesian coordinates (R, G, B). One can also consider the color space to be a 3D Riemannian manifold [22].
The main point in this example is that the notion of color is space independent. It means that our definition of redness, greenness and blueness is THE SAME on each and every point on the surface. In other words if we define unit vectors in the R, G and B directions, say, $\mathbf{r}$, $\mathbf{g}$ and $\mathbf{b}$ respectively then a color is represented as $C(x,y) = R(x,y)\mathbf{r} + G(x,y)\mathbf{g} + B(x,y)\mathbf{b}$. Note that the unit vectors are CONSTANTS with respect to x and y. This situation when the basis vectors themselves are spatially dependent occurs often in vision problem. We defer the treatment of this problem to future publication.
Here we describe the color image on a surface as embedding of a 2D surface S embedded in 5D manifold $\mathcal{M} = \Sigma \times \mathbb{R}^3$. The metric in the embedding space is

$$(h_{ij}) = \begin{pmatrix} \tilde{g}_{11} & \tilde{g}_{12} & 0 & 0 & 0 \\ \tilde{g}_{21} & \tilde{g}_{22} & 0 & 0 & 0 \\ 0 & 0 & \beta^2 & 0 & 0 \\ 0 & 0 & 0 & \beta^2 & 0 \\ 0 & 0 & 0 & 0 & \beta^2 \end{pmatrix}.$$

The line element is therefore

$$ds_{\mathcal{M}} = \tilde{g}_{11}dx^2 + 2\tilde{g}_{12}dxdy + \tilde{g}_{22}dy^2 + \beta^2\left(dR^2 + dG^2 + dB^2\right).$$

and the induced metric is

$$G = (g_{ij}) = \begin{pmatrix} \tilde{g}_{11} + \beta^2\left(R_x^2 + G_x^2 + B_x^2\right) & \tilde{g}_{12} + \beta^2\left(R_xR_y + G_xG_y + B_xB_y\right) \\ \tilde{g}_{21} + \beta^2\left(R_xR_y + G_xG_y + B_xB_y\right) & \tilde{g}_{22} + \beta^2\left(R_y^2 + G_y^2 + B_y^2\right) \end{pmatrix}$$

It is easy to see that the Levi-Civita connection coefficients vanish in this case for the same reason it does in the scalar case. We finally obtain the equations of motion

$$U_t^a = \Delta_g U^a = \frac{1}{\sqrt{g}}\mathrm{Div}\left(\sqrt{g}G^{-1}\nabla U^a\right) \quad a = r, g, b$$

5.1 Example: Color image on the sphere

In order to apply the above equations to an image "painted" on the sphere we need the metric $\tilde{g}_{ij}$ of the sphere. We choose to represent the sphere by stereographic coordinates from the south pole. The metric in this case was calculated already in the scalar case and it is

$$\tilde{g}_{ij} = \frac{4}{1+x^2+y^2}\delta_{ij} \ .$$

The induced metric of the two-dimensional section of this five-dimensional fiber bundle is

$$G = (g_{ij}) = \left(\tilde{g}_{ij} + \beta^2 \sum_{a=R,G,B} I_i^a I_j^a \right) \ ,$$

where $I_i^a = \partial U^a / \partial x^i$ with $x^1 = x$ and $x^2 = y$. Define

$$A(x,y) = \frac{4}{1+x^2+y^2} \ .$$

With this notation the determinant read

$$g = det(g_{ij}) = A^2 + A\beta^2 \sum_a ((U_x^a)^2 + (U_y^a)^2) + \beta^4 \sum_{ab} |\nabla U^a \times \nabla U^b|^2 \ .$$

The equation of motions are

$$U_t^a = \frac{1}{\sqrt{g}} \mathrm{Div} \left(\frac{1}{\sqrt{g}} \begin{pmatrix} A + \beta^2 \sum_b (U_y^b)^2 & -\beta^2 \sum_b U_x^b U_y^b \\ -\beta^2 \sum_b U_x^b U_y^b & A + \beta^2 \sum_b (U_x^b)^2 \end{pmatrix} \nabla U^a \right) \quad a = r, g, b.$$

6 The Beltrami flow over a curve: Implicit approach

In the previous sections, we formulated the Beltrami flow for images which are defined over non-flat surfaces that were given in an intrinsic-explicit parametric form. We would like to focus in the following section on the case where the surface is given in an implicit form. In the implicit setup we are given an implicit surface $\Psi(X^1, X^2, X^3) = 0$ and an implicit scalar image on the surface $U(X^1, X^2, X^3)$. We postpone the treatment of this case to the next Section. We wish to study first the easier case of a one-dimensional manifold i.e. a curve on which a scalar field is given. Although the derivation of the flow equations for scalar data defined over a curve and that over a surface is very similar it is easier to have a clear mental image of the situation for curves. We describe first, therefore, the derivation in details in this simpler

situation. The derivation in the first subsection is purely geometric, while we derive, in the second subsection the same equation from a variational point of view. It is this second, variational, approach that will be generalized in the next sections to higher dimensional manifolds.

6.1 Scalar field defined on a curve: Geometric derivation

The curve is given in the x-y plane and the data is pictured as the height in the z direction. The data from this point of view is a curve in $\mathbb{R}^3$. Note that this curve lies on the cylinder-like surface defined by $\Psi(x,y)=0$. We want to produce a curvature flow of this data such that it is confined to the surface which means that it is defined on the base curve. The projected curvature on the surface is the geodesic curvature. Note however that the fact that the data curve is on the cylinder-like surface, generated by the base curve, is not enough in general to guaranty that the data curve can be represented as a function on the base curve since the evolution is not done in the z direction only (compare with [5]). It may happen, for a general flow, that at some time of the flow more than one point of the data curve has the same x-y values. In order to avoid that and in order to have a flow that respect discontinuities we confine the flow by projecting the geodesic curvature on the z direction. It is the analog of the Beltrami flow which is a projection of the mean curvature flow on the intensity direction.
Formulating this geometric understanding, we represent the curve as the intersection of two surfaces :

$$\begin{aligned} \Psi(x,y) &= 0 \\ \Phi(x,y,z) &= z-\beta U(x,y)=0. \end{aligned} \tag{23}$$

The surface defined by Ψ is a cylinder-like surface defined by the base curve, which lies in the x-y plane. The second surface is the graph of the function U which is defined over the x-y plane (and by restriction over the base curve). The data function is modified along the flow in the z direction only. It remain, thus, a function along the flow (see Fig. 3). We denote by D the 3D gradient $D=(\partial_x,\partial_y,\partial_z)^t$, and by ∇ the 2D gradient $\nabla=(\partial_x,\partial_y)^t$. We define the tangent vector to the curve by

$$T=\frac{D\Psi\times D(z-\beta U)}{||D\Psi\times D(z-\beta U)||}=[T_1,T_2,T_3]^t$$

The curvature is given by the second derivative of the curve.

$$kN = \partial_s T = x_s\partial_x T + y_s\partial_y T + z_s\partial_z T = T^t DT$$

$$= \quad (T \cdot DT_1, T \cdot DT_2, T \cdot DT_3)^t \tag{24}$$

The projection of this vector on the surface defined by $\Psi(x, y) = 0$ is given by the projection operator applied to this vector – $P_\eta kN$. The normal to the surface is

$$\eta = \frac{D\Psi}{||D\Psi||} \ .$$

We will write, by abuse of notation, $P_{D\Psi}$ and $P_{\nabla\Psi}$ for the 3D and 2D projections respectively. Keeping only the z component in Eq. (24) we get

$$U_t = (P_{D\Psi} kN)_z \tag{25}$$

which is the non-flat analogue of Eq. (14).

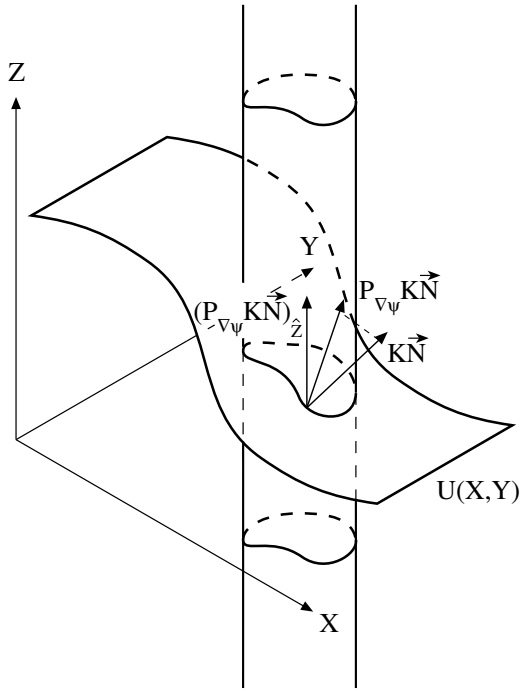


Figure 3: The data curve as the intersection of the cylinder-like surface induced by the base planar curve, extended to $\mathbb{R}^3$, and the graph of the function U.

Let us calculate it explicitly. We have the following:

$$\begin{aligned}(D\Psi \times D(z-\beta U))^t &= (\Psi_y, -\Psi_x, \beta(\Psi_y U_x - \Psi_x U_y)) \\ D(z-\beta U)^t &= (-\beta U_x, -\beta U_y, 1) \\ D\Psi^t &= (\Psi_x, \Psi_y, 0) \end{aligned} \tag{26}$$

Note that T does not depend on z! The projection operator is

$$\begin{aligned}P_{D\Psi} &= I - \frac{D\Psi D\Psi^t}{||D\Psi||^2} \\ &= \frac{1}{||D\Psi||^2}\begin{pmatrix} \Psi_y^2 & -\Psi_x\Psi_y & 0 \\ -\Psi_x\Psi_y & \Psi_x^2 & 0 \\ 0 & 0 & ||D\Psi||^2 \end{pmatrix}\end{aligned}$$

The equation of motion now reads

$$U_t = \left(P_{D\Psi}\begin{pmatrix} T\cdot DT_1 \\ T\cdot DT_2 \\ T\cdot DT_3 \end{pmatrix}\right)_z = T\cdot DT_3$$

Since T does not depend on z, only the x and y components enter the calculation:

$$U_t = T_1\partial_x T_3 + T_2\partial_y T_3 \tag{27}$$

The gradient notation ∇ and all other vector notations are from now on two-dimensional. We also denote the perpendicular gradient by $\nabla^\perp\Psi = (\Psi_y, -\Psi_x)^t$. Notice also that in the x-y plane we have the identities

$$\begin{aligned}||\nabla\Psi||^2 P_{\nabla\Psi}\nabla U &= (\Psi_y U_x - \Psi_x U_y)\nabla^\perp\Psi \\ ||\nabla\Psi||^2||P_{\nabla\Psi}\nabla U||^2 &= (\Psi_y U_x - \Psi_x U_y)^2\end{aligned} \tag{28}$$

Define

$$\begin{aligned}g &= ||\nabla\Psi||^2 + \beta^2(\Psi_y U_x - \Psi_x U_y)^2 \\ &= ||\nabla\Psi||^2\left(1+\beta^2||P_{\nabla\Psi}\nabla U||^2\right).\end{aligned} \tag{29}$$

We can finally write Eq. (27) as follows

$$\begin{aligned}U_t &= \frac{\nabla^\perp\Psi}{\sqrt{g}}\cdot\nabla\left(\frac{\beta(\Psi_y U_x - \Psi_x U_y)}{\sqrt{g}}\right) \\ &= \frac{1}{\sqrt{g}}\mathrm{Div}\left(\frac{\beta||\nabla\Psi||^2 P_{\nabla\Psi}\nabla U}{\sqrt{g}}\right)\end{aligned} \tag{30}$$

where we used the identities Eq. (28) and the fact that $\mathrm{Div}\nabla^{\perp}\Psi = \partial_x(\Psi_y) + \partial_y(-\Psi_x) = 0$.
In conclusion, this last equation allows us to denoise the scalar signal U lying on the implicitly defined curve $\Psi(x,y) = 0$. This Beltrami flow defined on implicit curve can also be generalized to the case of scalar field defined on implicit surface. In order to do that we re-derive the same equation, for the curve. The next subsection approach this problem from calculus of variations viewpoint.

6.2 Scalar field on a curve: Variational derivation

The variational formulation is simply the minimization of the **length of the data curve**. The curve is given, as before, by the intersection of two surfaces:

$$\begin{aligned} \Psi(x,y) &= 0 \\ \Phi(x,y,z) &= z - \beta U(x,y) = 0 \end{aligned} \tag{31}$$

The length of the curve is given by integrating the non-normalized tangent vector along the curve. In explicit-implicit formulation we would write $L = \int |C'(p)|dp$ where $C = (x(p), y(p), z(p))$. In the implicit formulation we note that the curve is given as the intersection of two surfaces. The tangent to the curve is, thus, perpendicular to the normals of the two surfaces. The length, in this case, takes the following form:

$$L = \int \delta(\Psi(x,y))\delta(\Phi(x,y,z))||D\Psi(xy) \times D\Phi(x,y,z)||dxdydz \ . \tag{32}$$

Yet we insist that Φ cannot be any function that obey the flow. It must also conserve the form $\Phi(x,y,z) = z - \beta U(x,y)$ **along the flow**. This means that we can plug this form in the integral and minimize directly with respect to $U(x,y)$. This operation is in fact the projection in the z direction. Our functional, therefore, is

$$\begin{aligned} L &= \int \delta(\Psi(x,y))\delta(z - \beta U(x,y))||D\Psi(xy) \times D(z - \beta U(x,y))||dxdydz \\ &= \int \delta(\Psi(x,y))\delta(z - \beta U(x,y))\sqrt{||D\Psi||^2 + \beta^2(\Psi_y U_x - \Psi_x U_y)^2}dxdydz \\ &= \int \delta(\Psi(x,y))\delta(z - \beta U(x,y))\sqrt{||\nabla\Psi||^2 + \beta^2(\Psi_y U_x - \Psi_x U_y)^2}dxdydz \\ &= \int \delta(\Psi(x,y))\sqrt{||\nabla\Psi||^2 + \beta^2(\Psi_y U_x - \Psi_x U_y)^2}dxdy \ . \end{aligned} \tag{33}$$

We used here the fact that Ψ does not depend on z to wrote $||D\Psi|| = ||\nabla\Psi||$ and since nothing depends anymore on z we can integrate it out, which leads to the last equality.
Define

$$g = ||\nabla\Psi||^2 + \beta^2(\Psi_y U_x - \Psi_x U_y)^2 \ .$$

The flow is simply the (modified) gradient descent equation

$$\begin{aligned} u_t &= -\frac{1}{\beta\sqrt{g}}\frac{\delta L}{\delta U} \\ &= \frac{1}{\sqrt{g}}\mathrm{Div}\left(\beta\frac{||\nabla\Psi||(\Psi_y u_x - \Psi_x u_y)\nabla^\perp\Psi}{\sqrt{g}}\right) \end{aligned} \tag{34}$$

and using the identities Eq. (28) we find the same flow Eq. (30) as in the previous section.

7 scalar field on an implicit manifold

We treat here the case of a scalar field i.e. gray-value image, on an n-Dimensional manifold. We describe the manifold by the zero level of a function on $\mathbb{R}^{n+1}$ i.e. $\Psi(x_1,\ldots,x_{n+1}) = 0$ and we extend the scalar field to $\mathbb{R}^{n+1}$ as well by the function $U(x_1,\ldots,x_{n+1})$. The data manifold is described then as an n-Dimensional manifold embedded in $\mathbb{R}^{n+2}$. The data manifold is given by the intersection of the two implicit $(n+1)D$ hyperplanes $\Psi(x_1,\ldots,x_{n+1}) = 0$ and $\Phi(x_1,\ldots,x_{n+1}) = x_{n+2} - \beta U(x_1,\ldots,x_{n+1}) = 0$. Note that Ψ does not depend on x_{n+2}. This is not the most general form because it is not always true that one can represent an n-Dimensional manifold as a zero set of a function in (n+1) Euclidean space (see Nash embedding theorem [14]). Nevertheless this embedding is always possible for 2d surfaces that we treat below. It is also possible in most cases of interest in computer vision and computer graphics. We define as above the $\mathbb{R}^{n+2}$ gradient D and the $\mathbb{R}^{n+1}$ gradient ∇. The functional is simply the volume of this n-Dimensional manifold. Note however that this is the **data manifold** and not the underlying fixed manifold described by Ψ.
The length functional, that was described in the previous section, is written in a manner that resist generalization. Let us first use Eq. (28) to rewrite it as follows:

$$L = \int \delta(\Psi(x,y,z))\delta(\Phi(x,y,z))||D\Psi \times D\Phi||dxdydz$$

$$= \int \delta(\Psi(x,y,z))\delta(\Phi(x,y,z))||D\Psi||||P_{D\Psi}D\Phi||dxdydz \ . \tag{35}$$

It is exactly this form that we generalize directly from curves to surfaces :

$$A = \int \delta(\Psi(x,y,z,w))\delta(\Phi(x,y,z,w))||D\Psi||||P_{D\Psi}D\Phi||dxdydzdw \ . \tag{36}$$

We claim that A, defined by Eq. (36), is exactly the area of the surface which is the intersection between the two 3-dimensional hyperplane, defined implicitly via functions in $\mathbb{R}^4$. It is quit easy to see, via simple examples, that A is indeed the area. We will give below two such examples. Yet, we couldn't find in the literature this formula, let alone its proof. We defer the proof of this fact (and its higher dimensional analogs) for a future publication. Let us look in the meantime at the following examples:

- **A scalar data over the 2D plane**: The image is given explicitly by the function $U(x,y)$ defined over $\mathbb{R}^2$. The graph of this function is our data surface. This surface is given, intrinsically, by the mapping $S = (x, y, z = \beta U(x,y))$. The area of the surface is given by

$$Area = \int dxdy\sqrt{g} = \int dxdy\sqrt{1+\beta^2(U_x^2+U_y^2)} \ .$$

  The implicit description of the surface is the intersection of the following implicit functions

$$\begin{aligned} \Psi(x,y,z,w) &= z \\ \Phi(x,y,z,w) &= w - \beta U(x,y,z) \end{aligned} \tag{37}$$

  We can calculate directly

$$\begin{aligned} ||D\Psi|| &= 1 \\ P_{D\Psi}D\Phi &= \left[\begin{pmatrix} 1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1 \end{pmatrix} - \begin{pmatrix} 0&0&0&0\\0&0&0&0\\0&0&1&0\\0&0&0&0 \end{pmatrix}\right]\begin{pmatrix} -\beta U_x\\-\beta U_y\\-\beta U_z\\1 \end{pmatrix} \end{aligned} \tag{38}$$

  Such that $||P_{D\Psi}D\Phi|| = \sqrt{1+\beta^2(U_x^2+U_y^2)} = \sqrt{1+\beta^2||\nabla U||^2}$. Note that this expression does not depend on z and w anymore. The integration over z and w, in the functional A, given in Eq. (36), can now be carried out trivially. We find, with no surprise, that A is exactly the area of the surface, as claimed.

- **The 2D sphere**: The intrinsic description of the sphere (for all, but the north pole, points) is $S = (x, y, z = \sqrt{1 - x^2 - y^2})$. The metric is

$$G = \begin{pmatrix} 1 + \frac{x^2}{1-x^2-y^2} & \frac{xy}{1-x^2-y^2} \\ \frac{xy}{1-x^2-y^2} & 1 + \frac{y^2}{1-x^2-y^2} \end{pmatrix}$$

and denoting $g = \det(G)$ we finally get

$$Area = \int dxdy\sqrt{g} = \int dxdy \frac{1}{\sqrt{1 - x^2 - y^2}} \ .$$

In the implicit approach the sphere is the intersection of zero level sets the following functions

$$\begin{aligned} \Psi(x, y, z, w) &= x^2 + y^2 + z^2 - 1 \\ \Phi(x, y, z, w) &= w - 1 \end{aligned} \tag{39}$$

It is straightforward to show that

$$\begin{aligned} ||D\Psi|| &= 2\sqrt{x^2 + y^2 + z^2} \\ P_{D\Psi} D\Phi &= (0, 0, 0, 1)^t \ . \end{aligned} \tag{40}$$

The functional now reads

$$\begin{aligned} A &= \int \delta(x^2 + y^2 + z^2 - 1)\delta(w - 1)2\sqrt{x^2 + y^2 + z^2}dxdydzdw \\ &= 2\int \delta(x^2 + y^2 + z^2 - 1)dxdydz \end{aligned} \tag{41}$$

We use the identity

$$\delta(f(z)) = \frac{\delta(z)}{f'(z)|_{z^*}} \ ,$$

where z^* is the root of $f(z)$, in order to rewrite the functional in the following form

$$A = 2\int \frac{\delta(z)}{2\sqrt{1 - x^2 - y^2}} dxdydz = \int \frac{dxdy}{\sqrt{1 - x^2 - y^2}} \ . \tag{42}$$

Again we see that A is the area, as claimed.

Going back now to a scalar image defined over implicitly given n-dimensional manifold we write the volume of the data manifold as follows:

$$V = \int \delta(\Psi(x_1,\ldots,x_{n+1}))\delta(x_{n+2}-\beta U(x,\ldots,x_{n+1}))||D\Psi||\;||P_{D\Psi}D(x_{n+2}-\beta U)||dx_1\cdots dx_{n+2}$$

Notice that in this setup we have

$$||P_{D\Psi}D(x_{n+2}-\beta U||^2 = (\quad -\beta\nabla U \quad , \quad 1\,)\begin{pmatrix} & & & 0 \\ & P_{\nabla\Psi} & & \vdots \\ & & & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}\begin{pmatrix} -\beta\nabla U \\ \\ 1 \end{pmatrix} = 1+\beta^2||P_{\nabla\Psi}\nabla U||^2$$

Using also the fact that $||D\Psi|| = ||\nabla\Psi||$ we get

$$\begin{aligned} A &= \int \delta(\Psi(x,\ldots,x_{n+1}))\delta(x_{n+2}-\beta U(x,\ldots,x_{n+1}))||\nabla\Psi||\sqrt{1+\beta^2||P_{\nabla\Psi}\nabla U||^2}dx\cdots dx_{n+2} \\ &= \int \delta(\Psi(x,\ldots,x_{n+1}))||\nabla\Psi||\sqrt{1+\beta^2||P_{\nabla\Psi}\nabla U||^2}dx\cdots dx_{n+1} \end{aligned} \tag{43}$$

The equation of motion is derived by a (modified) gradient descent equation

$$U_t = \frac{1}{||\nabla\Psi||\sqrt{1+\beta^2||P_{\nabla\Psi}\nabla U||^2}}\mathrm{Div}\left(\frac{\beta||\nabla\Psi||P_{\nabla\Psi}\nabla U}{\sqrt{1+\beta^2||P_{\nabla\Psi}\nabla U||^2}}\right) \tag{44}$$

This is the direct generalization of the flow we obtained in the previous section for a scalar data field defined over a curve.

8 The L_1 and L_2 limits

We show in this subsection that Eq. (44) interpolates between the non-flat L_1 flow Eq. (8) and the L_2 flow Eq. (5).
Note that

$$\begin{aligned} \lim_{\beta\to 0} g &= ||\nabla\Psi||^2 \\ \lim_{\beta\to\infty} g &= \beta^2||\nabla\Psi||^2||P_{\nabla\Psi}\nabla U||^2 \end{aligned} \tag{45}$$

It is easily seen now that the implicit Beltrami flow Eq. (44), in the limit $\beta \to 0, \quad t\to\infty$ such that $\tau = t\beta$ is finite, goes to

$$u_\tau = \frac{1}{||\nabla\Psi||}\mathrm{Div}\left(||\nabla\Psi||P_{\nabla\Psi}\nabla U\right),$$

which is the non-flat L_2 flow, which was derived differently by Bertalmio et al [1].
In the other limit, $\beta \to \infty, \quad t \to \infty$ such that $\tau = t/\beta$ is finite, we obtain another flow :

$$u_\tau = \frac{1}{||\nabla\Psi||||P_{\nabla\Psi}\nabla U||}\mathrm{Div}\left(\frac{||\nabla\Psi||P_{\nabla\Psi}\nabla U}{||P_{\nabla\Psi}\nabla U||}\right)$$

which is the non-flat L_1 flow, used in [1].
In one limit, we find the L_2 norm which over-smooth the image, while in the other limit we find the L_1 flow with the notorious "staircasing" effect. Choosing for β intermediate value brings more degree of freedom in the regularization of noisy data defined on surfaces. It opens news perspectives with the advantage of smoothing anisotropically the image and conserving the edges while avoiding the disadvantages of the total variation type of flow. This is well illustrated and demonstrated on various synthetic and real images in Section 10.

9 Bi-dimensional vector field on a surface: Implicit approach

We have in this case an N-dimensional manifold which is described by an implicit function on $\mathbb{R}^{n+1}$: $\Psi(x_1,\ldots,x_{n+1})$. On this manifold we have a bi-dimensional vector field with components U_1 and U_2 extended to $\mathbb{R}^{n+1}$ as well. We describe the **data manifold** as the intersection of the following functions:

$$\begin{aligned}
\Psi(x_1,\ldots,x_{n+1}) &= 0 \\
\Phi_1(x_1,\ldots,x_{n+3}) &= x_{n+2} - \beta U_1(x_1,\ldots,x_{n+1}) = 0 \\
\Phi_2(x_1,\ldots,x_{n+3}) &= x_{n+3} - \beta U_2(x_1,\ldots,x_{n+1}) = 0
\end{aligned} \tag{46}$$

The functional is the volume of the manifold of intersection of these three functions. It is given by

$$S[U_1,U_2] = \int \delta(\Psi)\delta(\Phi_1)\delta(\Phi_2)||D\Psi||||P_{D\Psi}D\Phi_1||||P_{P_{D\Psi}D\Phi_1}P_{D\Psi}D\Phi_2|| \tag{47}$$

In order to write it in a simpler form we use the following identity. Let v_1 and v_2 be two vectors. then

$$||v_1||^2||P_{v_1}v_2||^2 = ||v_1||^2||v_2||^2 - (v_1\cdot v_2)^2 \tag{48}$$

It follows that

$$||P_{D\Psi}D\Phi_1||^2||P_{P_{D\Psi}D\Phi_1}P_{D\Psi}D\Phi_2||^2 = ||P_{D\Psi}D\Phi_1||^2||P_{D\Psi}D\Phi_2||^2 - (P_{D\Psi}D\Phi_1\cdot P_{D\Psi}D\Phi_2)^2 \tag{49}$$

Next we recall our distinction between the gradient D in $\mathbb{R}^{n+3}$ and ∇ which is the gradient in the first $n+1$ coordinates. It follows that

$$\begin{aligned} P_{D\Psi} D\Phi_1 &= (-\beta P_{\nabla\Psi} \nabla U_1, 1, 0)^t \\ P_{D\Psi} D\Phi_2 &= (-\beta P_{\nabla\Psi} \nabla U_2, 0, 1)^t \ . \end{aligned} \tag{50}$$

Using these identities in Eq. (49) we arrive finally to a functional that reads

$$S[U_1, U_2] = \int \delta(\Psi) ||D\Psi|| \sqrt{1 + \beta^2 \sum_i ||P_{\nabla\psi} \nabla U_i||^2 + \beta^4 ||P_{\nabla\psi} \nabla U_1 \times P_{\nabla\psi} \nabla U_2||^2} \ . \tag{51}$$

This functional is the obvious generalization of the functional Eq. (15).

10 Examples and Results

In this section, we first give some implementation details and then illustrate the capabilities of the approach we have developed to regularize noisy signals and images defined on implicit curves and surfaces. Various experimental results have been carried out, we just give some figures for illustrations.

10.1 Implementing the regularization of scalar fields on surfaces

We compute the value of $u^n_{i,j,k}$, the value of u in the pixel (i,j,k) at the n^{th} iteration, based on the values of u^{n-1} at the neighboring pixels. First, we compute the vector $\vec{N} \simeq \nabla\Psi$ (this is needed only once), by central differences. Then, for each iteration n, we visit all pixels to compute:

- The gradient, its projection, and g.
- The divergence.
- The actualization of u.

For the gradient $\vec{v}^n_{i,j,k}$ we used forward differences,

$$\vec{v}^n_{i,j,k} = \nabla_+ u^n_{i,j,k} = \begin{pmatrix} u^n_{i+1,j,k} - u^n_{i,j,k} \\ u^n_{i,j+1,k} - u^n_{i,j,k} \\ u^n_{i,j,k+1} - u^n_{i,j,k} \end{pmatrix}$$

its projection on the surface,

$$\left(P_{\vec{N}}\vec{v}\right)^n_{i,j,k} = \vec{v}^n_{i,j,k} - \frac{\sum_{m=1}^{3} \vec{N}_{i,j,k}[m] \cdot \vec{v}^n_{i,j,k}[m]}{\left\|\vec{N}_{i,j,k}\right\|^2} \vec{N}_{i,j,k}$$

and g,

$$g^n_{i,j,k} = ||\vec{N}_{i,j,k}||^2 \left(1 + \beta^2 || \left(P_{\vec{N}}\vec{v}\right)^n_{i,j,k} ||^2\right),$$

where square brackets represent the component of the vector. To compute the divergence, we use backward differences,

$$\begin{aligned} \nabla_- \cdot \vec{w}_{i,j,k} \quad = \quad & \vec{w}_{i,j,k}[1] - \vec{w}_{i-1,j,k}[1] + \\ & \vec{w}_{i,j,k}[2] - \vec{w}_{i,j-1,k}[2] + \\ & \vec{w}_{i,j,k}[3] - \vec{w}_{i,j,k-1}[3] \end{aligned}$$

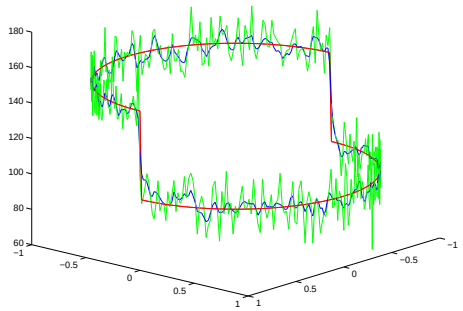
We switch forward differences and backward differences to avoid numerical problems. Finally, the flow implementation is:

$$u^{n+1}_{i,j,k} = u^n_{i,j,k} + \Delta t \frac{1}{\sqrt{g^n_{i,j,k}}} \nabla_- \cdot \left(\frac{\beta \left\|\vec{N}_{i,j,k}\right\|^2 \left(P_{\vec{N}}\vec{v}\right)^n_{i,j,k}}{\sqrt{g^n_{i,j,k}}} \right)$$

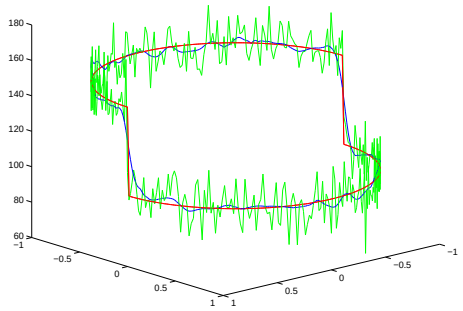
We use a time step $\frac{dt}{\beta}$, adjusted accordingly to section 3.3.
The code is made in C++ using the libraries developed in our lab. For the visualization, we used the marching cubes algorithm [12] to obtain a triangulation from our implicit representation of the surface, and draw the data on this surface. This was made using VTK.

10.2 Examples

We present in this subsection few figures that illustrate the regularization of noisy data on various implicitly defined curves and surfaces. The results are given with various values of the parameter β. Note how the regularization of the data is done isotropically or anisotropically depending on the value of this parameter.
For the next set of images: the red curve is the original and the blue curve is after regularization. For images (a) and (b), we added a Gaussian noise with $\sigma = 15$.



(a) $\beta = 0.1$



(b) $\beta = 0$ (Isotropic diffusion)

Regarding the CPU time, for the tori (12500 pixels), it took less than 2 seconds for 20 iterations. Please note that we used a color map from red to blue in order to better see how the discontinuities are treated. Regarding the noise level, we have added a Gaussian noise with ($\sigma = 40$) to the grey level images which scalar range is (0,256).

For the Ella image, illustrated in Fig. (5) and defined in a solid that looks like a quadratic, the image is much bigger (2744000 pixels). The time to compute 20 iterations was almost 3.5 minutes in a 386 sun, 260 MB in RAM. We can see that in strongly noised images like these, where anisotropic regularization treat some noise as part of the image discontinuities (the points under the left eye, for example) the regularization with $\beta = 0.1$ performs better.

Finally, for a slice from a cortex (97x222x143) acquired at IRMf Center of Marseille (N. Wotawa together with J.L. Anton and his colleagues are gratefully acknowledged for their support) it took less than 10 minutes to compute these results (fig 6). This is a retinotopic map i.e a neural representation within the visual cortex that preserves the spatial layout of the retina image. Notice the red holes in the noisy image. The

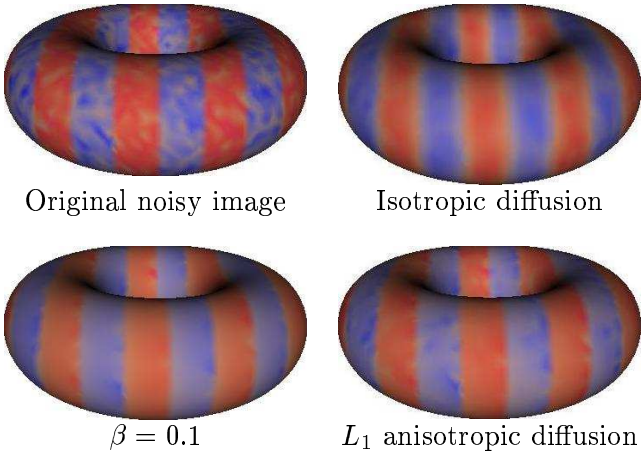


Figure 4: Synthetic image on the torus regularized with different β

small holes have to be filled in the map. Our approach allows to fill in these holes while preserving the important borders. While the isotropic smoothing also performs this task, it doesn't allow to preserve important information like the blue zone in the extreme right. With the L_1 anisotropic smoothing á la Bertalmio et al, the outer borders are thinner, but the inner holes rest. Our approach allows us to deal with more degree of freedom in the process to manage the weighting between the isotropy and anisotropy processes. Choosing beta in a range from 0.1 to 0.5 allows to produce a whole set of interesting and better results since the blue zone is kept and the holes are filled in.

11 Summary and conclusions

In this paper, we have first clarified the link that exists between the the intrinsic Polyakov action of the Beltrami framework and the implicit harmonic energy functional . It is found that although the functionals are basically the same, there are differences in the way various problems are formulated and consequently in the way the functionals are applied.
We used the geometrical understanding of the flat Beltrami flow and generalized it to a denoising flow over implicitly defined curves and surfaces. The same result was derived by a minimization of a functional. This functional was generalized first to

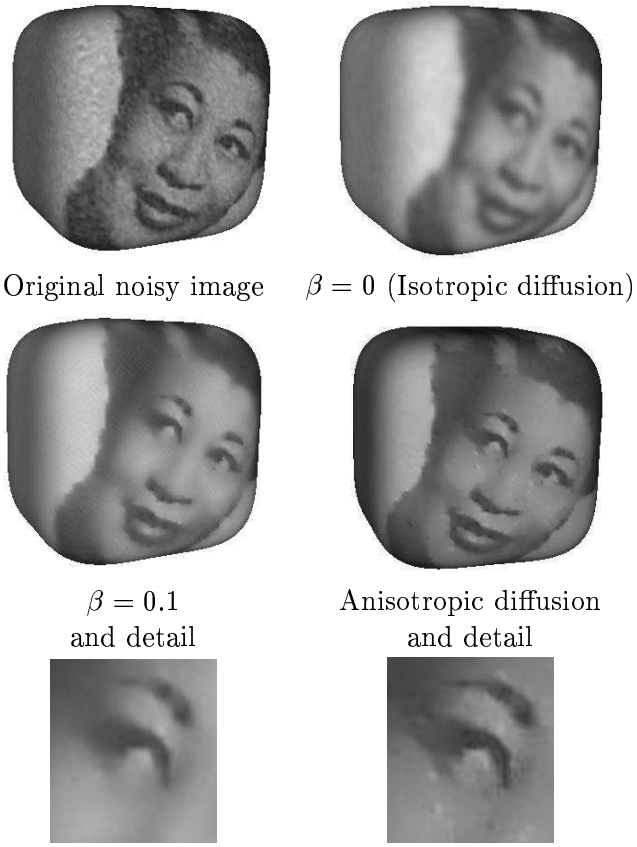


Figure 5: Ella image regularized with different β

a scalar field over an n-dimensional manifold. It was further generalized to treat vector value fields over an n-dimensional manifold. It is shown that the resulting flows depend on β which encodes the ratio between the data and spatial units. This parameter controls the edge-preserving characteristic of the flow. The Beltrami flow is shown to act as the linear diffusion (L_2-norm) in the limit $\beta \to 0$ and to the strongly edge preserving diffusion (L_1-norm for the scalar field) in the limit $\beta \to \infty$ (up to time scaling).

This work opens interesting perspectives in some important applications such as the inverse EEG-MEG problem. See for instance the work presented in [6] where the

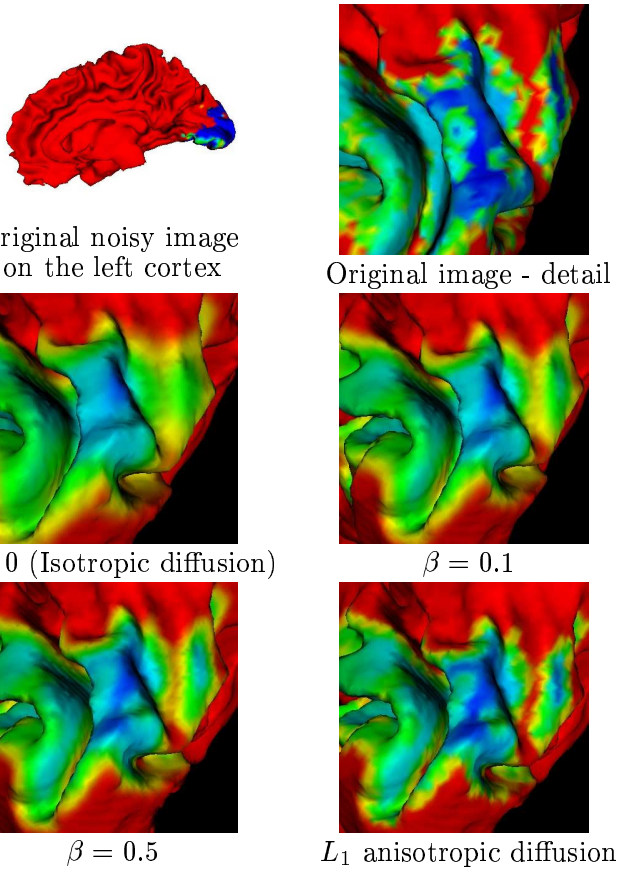


Figure 6: Retinotopic images regularized with different β

authors are interested by localizing cortex activity from EEG/MEG measurements. The PDE associated to the inverse problem includes a regularization term on the implicitly defined surface of the cortex. It could certainly be of interest to apply the implicit Beltrami flow, we developed, to regularize such data defined on implicit cortex surface. It will also be of great interest to compare our result to the methods developed to regularize data defined on triangulated surfaces.

Acknowledgments

Nir Sochen would like to thank the Odyssée project for its kind invitation to INRIA Sophia-Antipolis in summer 2002. N. Wotawa together with J.L. Anton and his colleagues from IRMf Center of Marseille are gratefully acknowledged for providing the cortex data. This research has been supported in part by the Israel Academy of Science, Tel-Aviv University fund, the Adams Super-Center for Brain Studies, the Israeli Ministry of Science, the European project MAPAWAMO, the INRIA Department of the European and International Relations (DREI) and the program CEFI-SFERE/CONACYT.

References

[1] M. Bertalmío, L. T. Cheng, S. Osher and G. Sapiro, "Variational Problems and Partial Differential Equations on Implicit Surfaces", Journal of Computational Physics 174 (2001) 759-780.

[2] T. Chan and J. Shen, "Variational restoration of non-flat image features: Models and algorithms", SIAM J. Appl. Math., 61 (2000) 1338-1361.

[3] C. Chefd'hotel, D. Tchumperle, R. Deriche, O. Faugeras, "Constrained Flows of Matrix-Valued Functions: Application to Diffusion Tensor Regularization", *Proceedings of 7th European Conference on Computer Vision*, Copenhagen, Denmark, May 2002

[4] C. Chefd'hotel, D. Tchumperle, R. Deriche, O. Faugeras, "Regularizing Flows for Constrained Matrix-Valued Images", *Journal of Mathematical Imaging and Vision* , Accepted for Publication, To appear in 2003

[5] L. Cheng, P. Burchard, B. Merriman and S. Osher, "Motion of Curves Constrained on Surfaces Using a Level Set Approach", September 2000 UCLA CAM Technical Report (00-32).

[6] M. Clerc, O. Faugeras, R. Keriven, J. Kybic, T. Papadopoulo, "A level set method for the inverse EEG/MEG problem", Poster No.: 10133, NeuroImage Human Brain Mapping 2002 Meeting

[7] R. Deriche and O. Faugeras, "Les EDP en Traitement des Images et Vision par Ordinateur", INRIA Research Report 2697, Nov. 1995. Also appeared in Traitement du Signal, 13 (6) 1996.

[8] R. Kimmel, "Intrinsic Scale Space for Images on Surfaces: The Geodesic Curvature Flow", *Graphical Models and Image Processing* 59(5) 365-372 1997.

[9] R. Kimmel and R. Malladi and N. Sochen, "Images as Embedding Maps and Minimal Surfaces: Movies, Color, Texture, and Volumetric Medical Images", International Journal of Computer Vision 39(2) (2000) 111-129.

[10] N. Sochen and R. Kimmel, "Stereographic Orientation Diffusion", in proceedings of the 4th Int. Conf. on Scale-Space, Vancouver Canada, October 2001.

[11] P. Kornprobst, R. Deriche, and G. Aubert. "Nonlinear operators in image restoration". In Proceedings of the International Conference on Computer Vision and Pattern Recognition - pages 325-331.IEEE Computer Society. San Juan, Puerto Rico, June 1997.

[12] W. E. Lorensen, H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm", Computer Graphics, 21(4), pages 163–169, 1987.

[13] F. Memoli and G. Sapiro and S. Osher, "Solving Variational Problems and Partial Differential Equations, Mapping into General Target Manifolds", January 2002 UCLA CAM Technical Report (02-04).

[14] John Nash, "The imbedding problem for Riemannian manifolds", Annals of Mathematics, 63 (1965), pp 20-63.

[15] A. M. Polyakov, "Quantum geometry of bosonic strings", *Physics Letters*, **103B** (1981) 207-210.

[16] L. Rudin, S. Osher and E. Fatemi, " Non Linear Total Variation Based Noise Removal Algorithms", *Physica D 60 (1992) 259-268.*

[17] G. Sapiro, "Geometric Partial Differential Equations and Image Analysis", Cambridge University Press, January 2001.

[18] R. Kimmel and N. Sochen, "Orientation Diffusion or How to comb a Porcupine", *Journal of Visual Communication and Image Representation* 13:238-248, 2001.

[19] N. Sochen and R. Kimmel and R. Malladi, "From high energy physics to low level vision", Report, LBNL, UC Berkeley, LBNL 39243, August, Presented in ONR workshop, UCLA, Sept. 5 1996.

[20] N. Sochen and R. Kimmel and R. Malladi, "A general framework for low level vision", *IEEE Trans. on Image Processing*, 7 (1998) 310-318.

[21] N. Sochen and Y. Y. Zeevi, "Representation of images by surfaces embedded in higher dimensional non-Euclidean space", 4th International Conference on Mathematical Methods for Curves and Surfaces, Lillehamer, Norway, July 1997.

[22] N. Sochen and Y. Y. Zeevi, "Representation of colored images by manifolds embedded in higher dimensional non-Euclidean space", Proc. IEEE ICIP'98, Chicago, 1998.

[23] B. Tang and G. Sapiro and V. Caselles, "Diffusion of General Data on Non-Flat Manifold via Harmonic Map Theory: The Direction Diffusion Case", *International Journal on Computer Vision* 36(2) 149-161, 2000.

[24] D. Tchumperlé and R. Deriche, "Orthonormal Vector Sets Regularization with PDEs and Applications", *International Journal on Computer Vision* 50(3) 237-252 (1992).

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399